

**Amendments to the Claims:**

This listing of claims will replace all prior versions, and listings, of claims in the application:

**Listing of Claims:**

1. (Currently Amended) A method of transmitting an INAP (Intelligent Network Application Protocol) in an IN (Intellectual Network), comprising:
  - generating an INAP message object indicating an INAP initial setting through an INAP factory object;
  - obtaining an invoke ID and dialogue ID from a TCAP (Transaction Capabilities Application Part) dialogue object, to set them in the INAP message object;
  - sending the INAP message object through the TCAP dialogue object such that the INAP message object calls a transmission component function of the TCAP dialogue object using the INAP object itself as a parameter;
  - issuing a component addition command, to add the INAP message object to a TCAP message object; and
  - generating and executing different transmission TCAP events based on a dialogue state, to send the INAP message and delete the object after sending the message.

2. (Previously Presented) The method as claimed in claim 1, wherein generating and executing different transmission TCAP events based on the current TCAP dialogue state to send and delete the INAP message object includes:

    inheriting the INAP object from a TC component object, to make a TC primitive object into TC beginning in a TCAP dialogue idle state;

    obtaining information required for the IC beginning from an SCP relationship object, to set the information in the TC primitive;

    issuing a TC primitive addition command to add the TC primitive to the TCAP message to which the TC component was added previously and to send it; and

    giving a command of transferring to an initial state, to transfer the TCAP dialogue object from the idle state into the initial state.

3. (Previously Presented) The method as claimed in claim 2, wherein issuing a TC primitive addition command to add the TC primitive to the TCAP message to which the TC component was added previously and to send it includes:

    processing the TC component into a format capable of being used in the TCAP or its lower network protocol;

    loading the processed TC component in a message data region, which is going to be transmitted to the TCAP, to send it to the TCAP; and

sending the TC component to the TCAP, and then transmitting the TC primitive of the TCAP message.

4. (Previously Presented) The method as claimed in claim 3, wherein transmitting the TC primitive of the TCAP message includes:

processing the TC primitive into a format capable of being used in the TCAP and its lower network protocol; and

loading the processed TC primitive in a message data region, which is going to be transmitted to the TCAP, to send it to the TCAP.

5. (Previously Presented) A method of receiving an INAP (Intelligent Network Application Protocol) in an IN (Intellectual Network), comprising:

when a TC primitive is received from a TCAP (Transactions Capabilities Application Part), generating a new TC beginning object for processing the received TC primitive;

finding a corresponding SCP (Service Control Point) relationship object using a dialogue ID included in the received TC primitive;

adding the new TC beginning object to a TCAP message by using the SCP relationship object;

when only the TC primitive is received without accompanying a TC component, executing the TC primitive to which the TC beginning object was added; and performing a corresponding call processing, and then deleting the SCP relationship object to finish a dialogue with the SCP,

wherein executing the TC primitive to which the TC beginning object was added includes:

when only the TC primitive is received, executing the TC primitive;  
when a TC component is received together with the TC primitive, processing the TC component while withholding the execution of the TC primitive; and carrying out the withheld TC primitive, and then executing the processed TC component.

6. (Previously Presented) The method as claimed in claim 5, wherein generating the TC beginning object includes generating a same TC primitive as a type of a received INAP message and decoding the received INAP message.

7. (Previously Presented) The method, as claimed in claim 5, wherein finding the corresponding SCP relationship object using a dialogue ID included in the received TC

primitive includes generating a new SCP relationship object through a TCAP dialogue object when the dialogue ID was not found.

8. (Canceled).

9. (Previously Presented) The method as claimed in claim 5, wherein processing the TC component includes:

when the TC component is received from the TCAP, analyzing an operation code loaded in a TC invoke object included in the received TC primitive;

generating a connection related INAP object when the operation code corresponds to an INAP connection related operation code;

executing the INAP object to find a corresponding SCP relationship object using a dialogue ID included in the received TC component;

commanding the SCP relationship object to add a TC component object, to add the TC component object to the TCAP message; and

executing the TC component.

10. (Previously Presented) The method as claimed in claim 5, wherein executing the TC primitive includes:

generating a TC primitive event indicating TC beginning, and changing the TCAP dialogue state corresponding to the idle state;

finding a corresponding SCP using address information of the TC beginning object, and then changing the TCAP dialogue state into a full duplex state;

when a TC component is included in the TCAP message, executing the TC component; and

ending a TCAP dialogue object when the TCAP dialogue state is changed into an ending state, and finishing a dialogue with the SCP.

11. (Previously Presented) The method as claimed in claim 5, wherein executing the TC primitive includes:

generating a TC primitive event indicating TC beginning, and changing the TCAP dialogue state corresponding to the idle state;

finding a corresponding SCP using address information of the TC beginning object, and then changing the TCAP dialogue state into the full duplex state;

when a TC component is included in the TCAP message, executing the TC component; and

ending the TCAP dialogue object when the TCAP dialogue state is changed into the ending state, and finishing the dialogue with the SCP.

12. (Previously Presented) The method as claimed in claim 5, wherein executing the TC component includes:

executing a component execution function, to inherit the TC component object and connection related INAP object;

reporting generation of a TC invoke event through a SCP relationship object;

returning the current call processing state to a TC invoke related INAP object for the reported event; and

performing a corresponding call processing according to the TC invoke event with respect to the current call processing state.

13. (Previously Presented) The method as claimed in claim 5, wherein ending the dialogue with the SCP includes:

calling a dialogue ending function, to generate a call processing event indicating conclusion of a SCP relationship;

reporting the generated SCP relationship ending event to a call processing object such that the current call processing state is returned; and

deleting the SCP relationship ending event, and then deleting the corresponding SCP relationship object to finish the dialogue with the SCP.

14. (Previously Presented) A method of processing an INAP (Intelligent Network Application Protocol) for communication between a SSP (Service Switching Point) and TCAP (Transaction Capabilities Application Part), comprising:

- the SSP generating a TC component to be sent to an SCP;
- allocating an invoke ID and dialogue ID to the generated TC component;
- generating a TC primitive corresponding to a state of the allocated dialogue;
- sequentially encoding the generated TC component and TC primitive in formats suitable for the protocol, to send it to the TCAP through a TCAP interface block;
- receiving an INAP message sent from the TCAP;
- decoding the received INAP message based on its kind;
- finding a corresponding SCP relationship object using the dialogue ID contained in the decoded INAP message;

- when the TC primitive is included in the decoded INAP message, executing the TC primitive; and

- upon execution of corresponding call processing, deleting the found SCP relationship object, to finish a dialogue with the SCP,

- wherein executing the TC primitive includes:

- when the TC primitive is the TC beginning that indicates initiation of the dialogue, generating a TC primitive event representing the TC beginning, and changing a



state of the dialogue corresponding to the dialogue ID included in the decoded INAP message from the idle state to the initial state;

finding a corresponding SCP using address information of a TC beginning and then changing the TCAP dialogue state from the initial state into a full state; and

when the decoded INAP message contains a TC component, executing the TC component.

15. (Previously Presented) The method as claimed in claim 14, wherein generating the TC primitive includes:

generating the TC primitive corresponding to TC beginning that indicates beginning of the dialogue when the state of the allocated dialogue corresponds to an idle state;

receiving a routing address required for the generated TC primitive from the SCP relationship object and adding it to the TCAP message; and

changing the state of the allocated dialogue from the idle state to an initial state.

16. (Canceled)

17. (Previously Presented) The method as claimed in claim 14, wherein executing the TC component includes:

executing a command corresponding to the operation of the TC component, to carry out a corresponding call processing; and

upon execution of the TC component, changing the state of the dialogue corresponding to the dialogue ID contained in the decoded INAP message into an ending state and finishing the dialogue.